



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

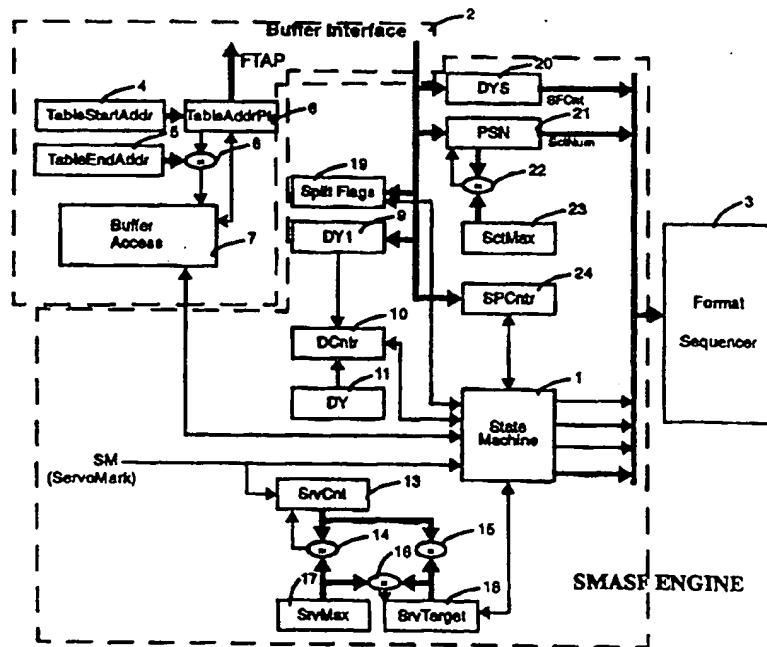
(51) International Patent Classification 6 : G11B 20/12, 20/10, 5/596, 27/10	A1	(11) International Publication Number: WO 95/24035 (43) International Publication Date: 8 September 1995 (08.09.95)
---	----	--

(21) International Application Number: PCT/US95/02608 (22) International Filing Date: 3 March 1995 (03.03.95) (30) Priority Data: 08/205,626 3 March 1994 (03.03.94) US (71) Applicant: CIRRUS LOGIC, INC. [US/US]; 3100 West Warren Avenue, Fremont, CA 94538-6423 (US). (72) Inventors: NEMAZIE, Siamack; 1253 Quail Creek Circle, San Jose, CA 95120 (US). HO, Son, H.; 972 Astor Court, Sunnyvale, CA 94086 (US). (74) Agent: IMAM, Maryam; Cirrus Logic, Inc., BS-906, 3100 West Warren Avenue, Fremont, CA 94538-6423 (US).	(81) Designated States: JP, KR, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.
--	---

(54) Title: A TABLE DRIVEN METHOD AND APPARATUS FOR AUTOMATIC SPLIT FIELD PROCESSING

(57) Abstract

A table driven method and apparatus for automatic split field processing in a disk drive system stores data representing the split fields after each servo mark of a frame. Each track of the disk drive system is divided up into frames or groups of sectors or equivalently servo marks, each frame comprising a predetermined number N of sectors and a predetermined number M of servo marks. Beginning from the INDEX mark, after every N sectors or equivalently M servo marks, the next sector must start after the next servo mark. The value N is a constant within each zone on a disk but can vary from zone to zone as the storage density changes. The information for each servo mark within a frame is stored in a frame table in the buffer memory (7). A starting address pointer (FTSAP) and an ending address pointer (FTEAP) are used to keep track of the starting and ending addresses, respectively, of the frame table. An address pointer (FTAP) is used to point to the current entry of interest in the frame table. For each servo mark within the frame, a sector pulse counter (24) is stored, representing the number of sector pulses between the current servo mark and the next servo mark. A first time delay DY1, representing the delay from the current servo mark to the first sector pulse to be generated and a second delay value DYS, representing the size of the last split field before the next servo mark are also stored.



BEST AVAILABLE COPY

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TC	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				

**A TABLE DRIVEN METHOD AND APPARATUS FOR
AUTOMATIC SPLIT FIELD PROCESSING**

FIELD OF THE INVENTION

5 The present invention relates to the control of storage systems for digital computers. More particularly, the present invention relates to the use of a storage table for automatically generating sector pulses and calculating split field sizes for control of storage in a disk drive system.

10

BACKGROUND OF THE INVENTION

An example of a magnetic media platter with embedded servo fields is illustrated in Figure 1. These servo fields may also be referred to as servo bursts or servo marks. Each concentric 15 circle schematically represents one track which is subdivided into multiple sectors and each radial line extending from the center outward, represents a servo field. A servo field contains data which is pre-written on the disk, during manufacturing of the disk, concerning the location of the servo field and the 20 track on the disk for positioning of the read/write head relative to a particular track. The servo fields are then used by the controller for positioning of the read/write head during read and write operations. It is imperative therefore, that the data within the servo fields not be written over. In order not to 25 write over the data within a servo field, the controller must know where the servo fields are and be able to suspend write operations over that area.

The usable regions for storage on the disk are located on the tracks, between the servo fields. Data to be stored on the 30 disk is written in these regions between the servo fields, which will be referred to as a data region. The recording of data within these regions is measured in bits per inch (BPI). Assuming a constant bit density to record a bit, the number of

bits which can be recorded between servo fields is much less for the inner circle, track 0, than the number of bits which can be recorded in the outermost circle, track N. Though possible, it is impractical to provide a unique recording and reading speed for each track. In order to take advantage of the higher storage potential of the outer tracks without exceeding the allowable density on the inner tracks, the disk is divided into multiple zones and a method of zone bit recording is used. The tracks are grouped into zones such that all the tracks in one zone are adjacent to each other. The data rate at which data is written to or read from the disk is constant for all tracks within a zone and different for each zone. Because the disk is rotated at the same speed for all the zones, in order to take advantage of the greater recording area of the outer zones, the recording speed is increased for the outer zones on the disk.

The data to be stored on a disk is broken up into manageable portions called sectors. Multiple sectors are generally stored on each track. An example of a typical format of a sector is illustrated in Figure 2a. Each sector includes an ID field, a GAP field, a data field and a PAD field. The ID field is the unique identification tag for each sector in a track of a disk drive media and differentiates one sector from another. Within the ID field are four subfields, the ID PLO subfield, the ID AM subfield, the ID subfield and the CRC subfield. The ID PLO subfield is a multiple byte field of a known pattern which is used by the phase lock loop of the encoder/decoder (ENDEC) to synchronize to the incoming data stream. The ID AM subfield is one byte of a known pattern which is used by the drive controller to synchronize or align to the beginning of the ID subfield. This synchronization is necessary to align the disk controller to a data byte boundary, in order to differentiate each segment of data to a particular field. The ID subfield follows the ID AM subfield and contains the actual identification for the sector

and consists of multiple bytes used to specify the address of the sector on the disk drive. The number of bytes within the ID subfield is at the discretion of the manufacturer of the disk drive and is controlled by the format of the particular disk drive used. The number of bytes within the ID subfield is the same for every sector and can comprise a cylinder high byte, a cylinder low byte, a sector number byte, a head number byte and a byte or two for defect management. The ID subfield is then followed by a fixed number of error detection bytes in the CRC subfield which are used to detect any errors in the ID field. Some formats, referred to as ID_Less formats, do not use an ID field, but rather use a header subfield within the DATA field.

The GAP field is a fixed number of bytes which are used to separate the ID field from the DATA field. The DATA field includes a DATA subfield and an error correction code (ECC) subfield. The DATA subfield is the portion of the sector where the actual data, which is communicated between the host computer and the disk drive, is stored. The ECC subfield is a fixed number of bytes tagged on to the end of the DATA subfield which are used to detect and correct soft or hard errors on the media within the capability of the code. This is necessary in order to avoid the transfer of erroneous data to and from the host computer.

A typical ID_Less sector format is illustrated in Figure 2b. In ID_Less formats, the ID field is replaced by a header subfield within the DATA field, thus combining the identification data and the DATA field into one field and reducing the number of bits necessary for each sector. The header subfield can be brought under the protection of the same ECC field as the DATA field and therefore afforded the same protection as the DATA field. The CRC field associated with the header subfield can be decreased or eliminated, further reducing the overhead of the system and eliminating the hardware which generates the CRC field. In

ID_Less formats, all of the information in the sector header may be predetermined by reading a small number of other sector headers on the track, usually one. The sync field or PLO field in an ID_Less format can be reduced in size or the reliability of 5 the sync circuitry improved since the sync field will always occur immediately after a sector pulse. Because some automatic calculation methods of the prior art calculate the split field sizes before the data field sync detection, they may not fully support an ID_Less format.

10 Employing constant density recording on a media with embedded servo fields where the servo fields are prerecorded at fixed angular distances relative to each other, will usually result in a track format that periodically requires a sector to be split in order to fit in the data regions. In order to obtain 15 optimal or near optimal utilization of the space between the servo fields on a disk it may be necessary for the fields within a sector to be split in order to accommodate a servo burst. It is desirable to determine the fields that are split and the split field size automatically, without intervention by the 20 microprocessor controlling the disk drive. It is also desirable to generate a sector pulse automatically to signal the beginning of each sector.

The disk drive system performs three major operations: format, read and write. The disk drive can format the disk for 25 storing data in the format supported by the system and the disk. The disk drive system also performs write operations to the disk to store data on the disk transmitted from the host computer to the disk drive system and read operations from the disk to read data from the disk and transmit it to the host computer. The 30 disk is continuously rotating within the disk drive system, even as the read/write head moves between zones, which may cause the read/write head to cross over one or more servo fields as it travels from one track or zone to another track or zone. The

controller circuit knows the angular position of the read/write head, but as the read/write head changes tracks between different zones, the number of sectors between servo marks changes, because of the change in bit density between zones. The controller does

5 not know how to translate this angular position after a zone change into a known position on the track relative to the sectors and may be unable to determine the relative positioning of the sector pulses on the specific track. To re-orient itself on the disk, the hard disk controller will normally wait for the INDEX 10 mark and orient itself from the INDEX mark for the track that it is on. However, waiting for the INDEX mark during a read or write operation increases the latency in reaching the next desired sector. This delay adds to the seek latency of the disk drive system.

15 An automatic method for determining the fields that are split and the split field size is used by Cirrus Logic, Inc. of Fremont, California in Part Number CL-SH360. This method imposes some constraints on track format, specifically by only allowing the data subfield to be split and not allowing a split in the ID 20 field, the GAP field and the ECC subfield. The length of the data split fields is stored within the ID field during format operation and used during read and write operations. In order to avoid intervention by the microprocessor controlling the drive during the format operation, the ID fields of the track are 25 stored in a table in the drive buffer memory and read from the drive buffer memory during format operation. This method does not support automatic generation of the sector pulse.

Another automatic method for determining the fields that are split and the split field size is used by Adaptec, Inc. of 30 Milpitas, California in Part Number AIC-7110. This method is based on a timing signal to indicate the time when a sector servo field is present in order to suspend and resume the disk's format, read or write operations. This timing signal can be

generated by circuitry external to the drive controller to indicate the time when a servo field is present in order to suspend and resume the operation of the disk. This timing signal must always be asserted within the same byte in a given sector.

5 However, due to fluctuations in detecting the sector servo field, this method is not reliable.

The timing signal for the AIC-7110 can also be generated when an internal counter underflows. In this method constraints are also imposed on the track format by not allowing a split in

10 the ID field. An internal counter is loaded from a first-in first-out stack (FIFO) with a number representing the number of bytes from the data field SYNC byte to the next servo field. The counter is then decremented upon detection or writing of a SYNC byte. When this counter becomes zero, the timing signal is

15 generated and the counter is loaded from the FIFO again. Using this method, the counter can be loaded automatically from the ID field or by the microprocessor controlling the drive. If the counter is loaded by the microprocessor controlling the drive, frequent intervention is required by the microprocessor. The

20 AIC-7110 does not support automatically generating the sector pulse.

Another automatic method for calculating split fields and split field size is taught by Machado et al. in U.S. Patent No. 5,255,136 issued on October 19, 1993. This method also embeds

25 the split information within the ID field and therefore does not allow the ID field to be split. The split field information is stored in count bytes which are stored on a byte count stack within the sequencer and will be loaded into a byte counter of the sequencer to be used by the sequencer for sequencing data

30 sectors to the disk surface or reconstructing the data sectors into unbroken continuous byte sequences. This method also does not support automatic generation of sector pulses.

Another method proposed by Estakhri et al. in U.S. Patent Application Serial Number 07/904,804 filed on June 25, 1992, calculates the split counts and delay from the servo sector field to the sector pulse algorithmically without any track format constraints. This method is restricted because it requires a first table to find the delay from the servo mark to the first sector pulse after the servo mark and a second table to find the physical sector number of the first sector after the servo mark if the read, write or format operation is not starting from the INDEX mark. If the two tables are not used, this method must start from the INDEX mark at the beginning of each track. The requirement of starting the disk operation from the INDEX mark adds on the average, one half of a disk revolution to the seek latency for this method. This will increase the seek latency an average of 5.5 msec to 8.3 msec depending on the disk drive system. This is not an acceptable delay for most applications. Because seek latency is a parameter which is used to compare the performance of disk drives, it is desirable to implement a method for split field calculation which will help to minimize the seek latency of the disk drive system. To avoid the increased latency associated with starting the disk operation from the INDEX mark, the two tables referred to above are used.

This method taught by Estakhri et al. also requires a free running clock operating at the disk data rate from the encoder/decoder. Also within this method there are implicit timing restrictions used to calculate split field sizes before data field sync detection, which may be a problem in the ID_Less formats depending on the clock that is used for the state machine executing the algorithm. This method does support formats in which the data field comprises a data subfield and an ECC subfield but must be extended to support formats in which the data field consists of a header subfield, a data subfield and an ECC subfield.

What is needed is a method for calculating split field data and automatically generating a sector pulse. What is also needed is a method for storing split field information for a frame or track in memory. What is further needed is a system which also

5 eliminates the embedding of split field information in the ID field of the sector. What is still further needed is a system which supports various track format constraints such as the ID_Less format and other format constraints such as not splitting the ECC subfield or in the case of formats that use an ID field

10 for the sector, not splitting the ID field.

SUMMARY OF THE INVENTION

A table driven method and apparatus for automatic split field processing in a disk drive system stores data representing the split fields after each servo mark of a frame. Each track of

15 the disk drive system is divided up into frames or groups of sectors or equivalently servo marks, each frame comprising a predetermined number N of sectors and a predetermined number M of servo marks. Beginning from the INDEX mark, after every N

20 sectors or equivalently M servo marks, the next sector must start after the next servo mark. The value N is a constant within each zone on a disk but can vary from zone to zone as the storage density changes.

The information for each servo mark within a frame is stored

25 in a frame table in the buffer memory. A starting address pointer and an ending address pointer are used to keep track of the starting and ending addresses, respectively, of the frame table. An address pointer is used to point to the current entry of interest in the frame table.

30 For each servo mark within the frame, a sector pulse counter is stored, representing the number of sector pulses between the current servo mark and the next servo mark. A first time delay DY1, representing the delay from the current servo mark to the

first sector pulse to be generated and a second count value DYS, representing the size of the last split field before the next servo mark are also stored.

5 BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates an example of a magnetic media platter with embedded radial servo fields and circular tracks.

Figure 2a illustrates the fields and subfields of a typical sector.

10 Figure 2b illustrates a typical ID_Less sector format.

Figure 3 illustrates a Frame Table and an entry within the Frame Table representing the corresponding values for a servo mark within the frame.

15 Figure 4 illustrates a block diagram of a Sector Mark And Split Field Engine SMASF showing its inputs and outputs.

Figure 5 illustrates a schematic diagram of a Sector Mark And Split Field Engine SMASF.

Figure 6 illustrates a state diagram representing the operation of the state machine of the present invention.

20 Figure 7 illustrates two examples of sector pulse generation and split field calculation using the method and apparatus of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

25 The table driven method of the present invention is based on storing the split data field information for each servo mark within a frame or a track in the Buffer random access memory (RAM). A frame includes a predetermined number N of sectors and a predetermined number M of servo marks ranging from a portion of 30 a track to a whole track. Each frame within a zone will contain the same number N of sectors and the same number M of servo marks and will be formatted for storing data in the same manner. The frame method adds a periodicity to the operation of the system by

requiring that, starting from the INDEX mark, after every N sectors or equivalently M servo marks, the next sector must begin after the next servo mark, even if there is enough space left to have a split field. In order for the repetitive nature of the 5 frame to be advantageous, the frame must be a manageable number of sectors in that the Frame Table requires only a modest number of data bits and the Frame Table is relatively simple to calculate.

Using a frame system divides the track up into groups of 10 sectors, wherein each group contains the same number of sectors and servo marks as every other group within a given zone. The calculations for the generation of sector pulses and split fields for each frame are the same. In order to store the sector pulse and split field information for the whole zone in memory, using a 15 frame, all that is required is that this information be stored for one frame. The information is then repeated at the beginning of each new frame. Using a frame method may cause some space on the track to be wasted after the last sector of the frame, before the next servo mark, because the next sector must start after the 20 next servo mark.

A Frame Table, illustrated in Figure 3, containing information for each servo mark within the frame, is stored in the Buffer RAM. A six-byte entry or record of information is stored in the Frame Table for each servo mark in the frame. The 25 entry in the Frame Table for each servo mark includes a four-bit sector pulse counter value SPCnt, three split field flags, a relative physical sector number value rPSN, a fourteen-bit clock delay value DY1 and a fourteen-bit formatter's split byte count value DYS.

30 The four-bit sector pulse counter SPCnt is stored for each servo mark. This sector pulse counter SPCnt indicates the number of sector pulses to be generated between the current servo mark and the next servo mark. A fourteen-bit delay register DY1 is

also stored for each servo mark. The register DY1 indicates the clock delay from the servo mark to the first sector pulse to be generated after the servo mark. A fourteen-bit formatter's split byte count DYS is also stored for each servo mark. The register

5 DYS indicates the byte count of the last split field, measured from the sync byte of the split field until the end of the split field, which occurs at the next servo mark. A seven-bit relative physical sector number register rPSN is also stored for each servo mark. The register rPSN indicates the relative physical 10 sector number of the first sector after the current servo mark, relative to the start of the frame. This field rPSN can therefore range from zero for the first sector in the frame to N-1 for the last sector in the frame.

The three split field flags include a Split_ID flag, a 15 Split_GAP flag and a Split_DATA flag each signalling that their respective fields have been split. If the Split_ID flag is set, this then signals to the Sector Mark And Split Field Engine SMASF that the ID field has been split and that the remainder of the split sector beginning with the remainder of the ID field will 20 occur after the next servo mark. If the Split_GAP flag is set, this signals to the Sector Mark And Split Field Engine SMASF that the GAP field has been split and that the remainder of the split sector beginning with the DATA field will occur after the next servo mark. If the Split_DATA flag is set, this signals to the 25 Sector Mark And Split Field Engine SMASF that the DATA field has been split and that the remainder of the split sector beginning with the remainder of the DATA field will occur after the next servo mark.

In the preferred embodiment, the Frame Table must be 30 contiguous within the buffer RAM and can be located in any 64 Kbyte segment of the buffer. A Frame Table Start Address Pointer FTSAP points to a starting address of the Frame Table within the buffer memory. A Frame Table End Address Pointer FTEAP points to

an ending address of the Frame Table within the buffer memory. The Frame Table Start Address Pointer FTSAP and the Frame Table End Address Pointer FTEAP are both stored in sixteen-bit registers. A three-byte Frame Table Address Pointer FTAP points 5 to an entry in the Frame Table which represents an entry in the table. When the lower sixteen bits of the address pointer FTAP match the ending address pointer FTEAP, the lower sixteen bits of the address pointer FTAP is automatically reloaded with the starting address pointer FTSAP. The upper 8 bits of the address 10 pointer FTAP is a segment selector which selects a 64 Kbyte segment within the buffer RAM.

In the section of the buffer RAM where the Frame Table is stored, bit seven of the buffer data can be configured to be the odd parity of bits zero through six if a parity configuration bit 15 is set. If the parity configuration bit is not set, then bit seven of the buffer data is a don't care value and is internally reset.

A block diagram of a Sector Mark And Split Field Engine SMASF showing its inputs and outputs is illustrated in Figure 4. 20 The Sector Mark and Split Field Engine SMASF includes a clock input, a start input, a parity enable input, a servo mark input, a fetch done input, a buffer data bus input, and a buffer data bus strobe input. A fetch error and parity error signal, a buffer fetch signal, a sector signal, a load split field count 25 signal, an ID field split flag, a GAP field split flag, a DATA field split flag, a sector byte remaining count flag Use_SBRC, a physical sector number bus and a split field count bus are all output from the Sector Mark and Split Field Engine SMASF. The buffer data bus BD carries the buffer data as fetched from the 30 Frame Table. The buffer data bus strobe BD_stb is used to strobe the record from the Frame Table into the storage registers inside the Sector Mark and Split Field Engine SMASF.

A schematic block diagram of the Sector Mark And Split Field Engine SMASF is illustrated in Figure 5. The State Machine 1 in conjunction with the Format Sequencer 3 controls the operation of the Sector Mark And Split Field Engine SMASF which keeps track of 5 sector pulse timing between the servo marks with various delay counters and internal registers. In the preferred embodiment, the Sector Mark And Split Field Engine SMASF, the delay counters and the internal registers are controlled by a free-running clock which can be changed for different zones. If the clock is 10 changed, the Sector Mark And Split Field Engine SMASF is stopped during a zone change until the clock speed has been changed to the appropriate clock speed for that zone.

The Sector Mark And Split Field Engine SMASF includes an internal sixteen-bit delay register DY 11 which contains the 15 number of clock cycles between two sector pulses if they occur before the next servo mark. The delay value from the register DY 11 is used to generate the second and subsequent sector pulses after a servo mark. The first sector pulse is always generated using the value from the register DY1 9. Therefore, the register 20 DY 11 stores the number of clock cycles in a sector which is not split.

A sixteen-bit delay counter DCntr 10 is used by the Sector Mark And Split Field Engine SMASF to count down the delays corresponding to the delay values from the registers DY 11 and 25 DY1 9. An internal eight-bit physical sector number counter PSN 21 is used by the Sector Mark And Split Field Engine SMASF to keep track of the physical sector number of the sectors. The physical sector number counter PSN 21 is loadable by the drive microprocessor to initialize the number of the first sector to be 30 generated. The drive microprocessor can read the value of the relative physical sector mark number rPSN for the first sector after the current servo mark from the Frame Table entry for the target servo mark. After the physical sector number PSN is

initialized by the drive microprocessor, the onboard logic of the system then keeps track of the sector number by incrementing the physical sector number counter PSN 21 every time a sector pulse is generated. Before the system is initialized, the drive 5 microprocessor for the system detects the servo mark number from the information stored on the disk. This servo mark number is then translated by the microprocessor controller into a relative servo mark number corresponding to the servo mark number within the frame. To translate to the relative servo mark number the 10 microprocessor controller adds one to the detected servo mark number and then divides the result by the number M of servo marks within the frame. The result of this division operation will then include a quotient and a remainder value. The quotient is equal to the frame number on the track of the current frame and 15 the remainder value is equal to the relative servo mark number within the frame.

When the physical sector number counter PSN 21 is equal to the maximum sector number, stored in the register SctMax 23, the physical sector number counter PSN 21 then is reset to zero on 20 the next sector pulse. The eight-bit register SctMax 23 stores a value representing the number of the last sector on the track. The comparator 22 compares the physical sector number from the register PSN 21 to the maximum sector number from the register SctMax 23. A sector pulse counter SPCntr 24 keeps track of the 25 number of sector pulses which will be generated between the current servo mark and the next servo mark.

An eight-bit servo mark counter SrvCnt 13 is used by the Sector Mark And Split Field Engine SMASF to keep track of the servo mark number. The servo mark counter SrvCnt 13 is 30 incremented on the leading edge of a servo mark signal. When the servo mark counter SrvCnt 13 is equal to a maximum servo mark number, it is reset to zero on the next leading edge of the servo mark signal. The maximum servo mark number is stored in the

eight-bit register SrvMax 17 which stores a value representing the number of the last servo mark on the track. The comparator 14 compares the current servo mark number from the register SrvCnt 13 to the maximum servo mark number from the register 5 SrvMax 17. The comparator 18 compares the target servo mark number from the register SrvTarget 18 to the maximum servo mark number from the register SrvMax 17.

An eight-bit servo target register SrvTarget 18 is an internal register used by the Sector Mark And Split Field Engine 10 SMASF and stores the starting servo number for the servo mark on which the Sector Mark And Split Field Engine SMASF is to start its operation. When the Sector Mark And Split Field Engine SMASF starts, it will wait until the servo mark counter SrvCnt 13 is equal to the value in the servo target register SrvTarget 18 15 before starting its operation. While the Sector Mark And Split Field Engine SMASF is running, the servo target register will be incremented automatically before each subsequent servo mark is detected.

The Sector Mark And Split Field Engine SMASF also uses the 20 Frame Table pointers as described above. The sixteen-bit Frame Table Start Address Pointer FTSAP is stored in the register TableStartAddr 4 and points to the starting address of the Frame Table within the buffer RAM. The sixteen-bit Frame Table End Address Pointer FTEAP is stored in the register TableEndAddr 5 25 and points to the ending address of the Frame Table within the buffer RAM. The twentyfour-bit Frame Table Address Pointer FTAP is stored in the register TableAddrPtr 6 and points to the next record in the Frame Table through the Buffer Interface 2.

In order to manage the beginning of the operation, the drive 30 microprocessor, through the Buffer Manager, initializes the servo target register SrvTarget 18, initializes the Frame Table Address Pointer FTAP, stored in the register TableAddrPtr 6, to the entry in the Frame Table corresponding to the number of the servo mark

stored in the servo target register SrvTarget 18, and loads the physical sector number PSN 21 of the first sector after the servo target mark. The value stored in the servo target register SrvTarget 18 is initialized by the drive microprocessor and 5 contains the number of the servo mark at which the Sector Mark And Split Field Engine SMASF is to begin generating sector pulses. The number programmed into the servo target register SrvTarget 18 is a number of a servo mark which is a predetermined number of servo marks away from the current servo mark or 10 alternatively can be the servo mark number corresponding to the beginning of the next frame. If the servo target register SrvTarget 18 is to be programmed with a servo mark which is a predetermined number from the current servo mark, the target servo mark should be at least two servo marks ahead of the 15 current servo mark to allow the Sector Mark And Split Field Engine SMASF time to complete its initialization. It is possible that one servo mark ahead of the current servo mark can be enough to complete the initialization, but the preferred method of the present invention programs the target servo mark so that it is 20 two servo marks ahead of the current servo mark.

Once the drive microprocessor and the Buffer Interface 2 have performed the above initialization and the Sector Mark And Split Field Engine SMASF is started and has completed the fetch operation of the entry in the Frame Table corresponding to the 25 target servo mark, the Sector Mark And Split Field Engine SMASF then waits until the current servo mark is equal to the target servo mark before it begins generating sector pulses. The comparator 15 compares the current servo mark from the register SrvCnt 13 to the target servo mark from the register SrvTarget 30 18.

Once the servo mark counter SrvCnt 13 is equal to the value in the servo target register SrvTarget 18, the Sector Mark And Split Field Engine SMASF then checks again for the completion of

the fetch operation from the Frame Table in the buffer RAM. If the fetch operation has not been completed, the Sector Mark And Split Field Engine SMASF generates an error signal and the drive microprocessor then reinitializes the system, including

5 reprogramming the servo target register SrvTarget 18 with a new target servo mark. The checks for completion of the fetch operation are done for each servo mark.

Once the servo mark counter SrvCnt 13 is equal to the value stored in the servo target register SrvTarget 18 and the fetch 10 operation has been completed, then the Sector Mark And Split Field Engine SMASF loads the internal registers and counters with the information for the current servo mark from the Frame Table through the Buffer Interface 2. In the preferred embodiment of the invention the values retrieved by the fetch operation are 15 directly loaded into the internal registers. Alternatively, the values retrieved by the fetch operation can be loaded into holding registers before they are loaded into the internal registers. Once the values are loaded into the internal registers, the Sector Mark And Split Field Engine SMASF then 20 generates any sector pulses which are to be generated after the current servo mark, using the appropriate delay values and initializes the fetch operation for the values corresponding to the next servo mark.

The Format Sequencer 3 is separate from the Sector Mark And 25 Split Field Engine SMASF, performs all sector or track processing and keeps track of the split field counts, including which field has been split. The Sector Mark And Split Field Engine SMASF generates the sector pulses and split information. The Format Sequencer 3 includes a Sector Byte Remaining Counter SBRC for 30 keeping track of the byte count of the last split data field.

A state diagram representing the operation of the State Machine 1 of the present invention is illustrated in Figure 6. The State Machine 1 runs on a free running clock and includes the

ten states shown in Figure 6. The State Machine 1 begins in the Idle state 50 and stays there until a start instruction is received. Once the start instruction is received, the State Machine 1 then transitions to the Start state 52 and initiates 5 the buffer fetch operation, described above, to fetch the entry in the Frame Table corresponding to the target servo mark. The State Machine 1 then waits in the Start state 52 until it is instructed by the Buffer Interface 2 that the fetch operation has been completed.

10 Once the fetch operation is completed the State Machine 1 then goes into the SrvRise state 54 where it waits for the assertion of the servo mark signal. Once the assertion of the servo mark signal has been detected the State Machine 1 then goes to the SrvFall state 56 where it waits for the negation of the 15 servo mark signal, indicating that the servo field is complete.

After the negation of the servo mark signal has been detected, the State Machine 1 then goes into the SrvChk state 58 where several conditions are checked. The State Machine 1 at the state 58 first checks if the target servo mark has yet been 20 detected. If the current servo counter SrvCnt 13 is not equal to the target servo mark from the register SrvTarget 18 then the State Machine 1 goes back to the SrvRise state 54 to wait for the rising edge of the next servo mark.

If the current servo mark counter SrvCnt 13 is equal to the 25 value stored in the register SrvTarget 18 then the State Machine 1 at the state 58 next checks if the fetch operation has been completed. If the fetch operation has not been completed, this signals an error and the State Machine goes into the Error state 60. At the Error state 60, the State Machine 1 resets the Start 30 Flag to a logical low voltage level and sets an error flag. The State Machine 1 can also enter the Error state 60 if there is a parity error in bit 7 of the data in the Frame Table.

If the current servo mark counter SrvCnt 13 is equal to the value stored in the register SrvTarget 18, the fetch operation has been completed and there is no parity error, the State Machine 1 then checks to see if the sector pulse counter SPCnt 24 5 is equal to zero, signalling that no sector pulses are to be generated after this servo mark. If all the conditions above are true then the state machine automatically jumps to the LoadSplit_2 state 68 which will be discussed in detail below.

If the current servo mark counter SrvCnt 13 is equal to the 10 value stored in the register SrvTarget 18, the fetch operation has been completed, there is no parity error and the sector pulse counter SPCnt 24 is not equal to zero, the State Machine 1 then loads the delay counter DCntr 10 with the first sector pulse delay value DY1 and goes to the LoadSplit_1 state 62. At the 15 LoadSplit_1 state 62 the State Machine 1 initializes the Use_SBRC, ID_Split, GAP_Split and DATA_Split flags to the default values indicating that a new set of values for those flags will be loaded. In this case only the Use_SBRC flag is set because at least one more sector pulse is generated before the next servo 20 mark. Therefore, the Split Field Count value SFCnt should be ignored by the Format Sequencer 3 when the State Machine 1 asserts the signal LD_SFCNT.

After the State Machine 1 has initialized the flags at the LoadSplit_1 state 62, the State Machine 1 then goes to the w_DC 25 state 64 where the State Machine 1 decrements the delay counter until it reaches zero. This state furnishes a time delay for the next state which could be either the first sector pulse delay DY1 or the full sector delay DY. When the delay counter reaches zero, the State Machine then goes to the NxtSct state 66, where a 30 sector pulse is generated and loads the delay counter DCNTR 10. At the state 66 the value in the physical sector number register PSN 21 is incremented by one, if it is not equal to the maximum sector number from the register SctMax 23. If the sector pulse

counter SPCntr 24 value is then greater than one, the State Machine 1 will decrement the sector pulse counter SPCntr 24, load the delay counter DCntr 10 with the sector delay value DY from the register 11 and go back to the LoadSplit_1 state 62 for 5 initializing and waiting the delay time period for generating the next sector pulse.

Once the sector pulse counter SPCntr is equal to one, the State Machine 1 will decrement the sector pulse counter SPCntr 24, load the delay counter DCntr 10 with the sector delay value 10 from the register DY 11 and go to the LoadSplit_2 state 68. At this state 68 the State Machine 1 loads the split field count with the value from the DYS register 20, loads the split field flags with their respective values from the Frame Table and asserts the signal Ld_SFCnt as a strobe signal for the Format 15 Sequencer 3 to latch in the split field count and the output values. The Use_SBRC flag is set to a logical high voltage level if all of the split flags are equal to a logical low voltage level. The BufferFetch signal is asserted to initiate the fetch of the next entry in the Frame Table corresponding to the next 20 servo mark. Lastly, at the state 68 the target servo mark number SrvTarget is set to the value of the next servo mark number. This next value is either an increment of one if the target servo mark number SrvTarget is not equal to the maximum servo mark number from the register SrvMax 17 or is equal to zero if the 25 target servo mark number SrvTarget is equal to the maximum servo mark number. After the instructions for the state 68 are completed, the State Machine 1 goes back to the SrvRise state 54 to wait for the beginning of the next servo mark and repeats the process for that servo mark.

30 Two examples of sector pulse generation and split field calculation using the method and apparatus of the present invention are illustrated in Figure 7. In the first example, beginning with the servo mark 70, assuming that the system is

initialized and has completed the fetch for the servo mark 70, the Sector Mark And Split Field Engine SMASF then loads the delay counter 10 with the first sector pulse delay value from the delay register 9. Once the delay counter 10 has counted down from this 5 delay value, the Sector Mark And Split Field Engine SMASF then generates the sector pulse 82 and decrements the sector pulse counter 24. Because the sector pulse counter 24 is equal to a value greater than zero, signalling that at least one more sector pulse will be generated before the next servo mark, the Sector 10 Mark And Split Field Engine SMASF loads the delay counter 10 with the full sector pulse delay value from the delay register 11. Once the delay counter 10 has counted down from this delay value, the Sector Mark And Split Field Engine SMASF then generates the sector pulse 84 and decrements the sector pulse counter 24. 15 Because the sector pulse counter 24 is now equal to zero, signalling that no more sector pulses will be generated before the next servo mark 72, the Sector Mark And Split Field Engine SMASF loads the format sequencer with the split field count value DYS from the register 20.

20 The system then waits for the assertion of the next servo mark 72 and fetches the values from the frame table corresponding to that servo mark. After the assertion of the servo mark 72, the value of the sector pulse counter 24 is checked and because this value is greater than zero and a first sector pulse has not 25 yet been generated, the Sector Mark And Split Field Engine SMASF loads the delay counter 10 with the first sector pulse delay value from the register 9. Once the delay counter 10 has counted down from this delay value, the first sector pulse is generated and the sector pulse counter 10 is decremented. Because the 30 sector pulse counter 10 is now equal to zero, signalling that no more sector pulses are to be generated before the next servo mark, and all of the split flags are at a logical low voltage level, the Sector Mark And Split Field Engine SMASF recognizes

that this is the last sector within this frame and therefore it will not generate any more sector pulses until after the next servo mark 74.

In the second example of Figure 7, beginning with the servo 5 mark 76, a sector, beginning with the sector pulse 88, is illustrated which lasts more than two servo marks. The Sector Mark And Split Field Engine SMASF recognizes this situation because after the assertion of the servo mark 78, the sector pulse counter 24 is equal to zero, signalling that no sector 10 pulses will be generated after this servo mark 78. When the sector pulse counter 24 is equal to zero after the assertion of the servo mark 78, the Sector Mark And Split Field Engine SMASF automatically loads the format sequencer 3 with the split field 15 count value DYS from the register 20. The Sector Mark And Split Field Engine SMASF initiates the fetch of the next entry from the Frame Table and then waits until the assertion of the next servo mark 80. After the assertion of the servo mark 80, because the sector pulse counter is greater than zero, the delay counter 10 is loaded with the first sector pulse delay value DY1 from the 20 register 9. After the delay counter 10 has counted down from this delay value, the first sector pulse 90 is generated.

The number of sectors within a frame can be any number depending on the design and operation of a particular system. In the preferred embodiment of the present invention, at the 25 beginning of each track on the media within the disk drive system, a new frame is started. It is not required at the end of a track that the last frame is a full frame. The last frame on the track can be a partial frame, depending on the size of the track relative to the size of a frame, in order to maximize the 30 number of sectors which can fit on a track. Alternately, a system could be designed which requires the last frame on a track to be a full frame. In either of the above cases, the end of the track is detected, as explained above, by comparing the servo

target register Srv_Target 18 with a maximum servo mark number, stored in the register SrvMax 17. If the servo target register Srv_Target is equal to the maximum servo mark number, then the apparatus of the present invention is signalled that the end of 5 the track has been reached.

The present invention has been described in terms of specific embodiments incorporating details to facilitate the understanding of the principles of construction and operation of the invention. Such reference herein to specific embodiments and 10 details thereof is not intended to limit the scope of the claims appended hereto. It will be apparent to those skilled in the art that modifications may be made in the embodiment chosen for illustration without departing from the spirit and scope of the invention. Specifically, it will be apparent to one of ordinary skill in the art that the method of the present invention could be implemented in many different ways and the apparatus disclosed above is only illustrative of the preferred embodiment of the present invention.

C L A I M S

We Claim:

1. 1. A method of automatic split field calculation in a magnetic media system using a magnetic media having a first plurality of circular tracks, each track having a second plurality of radial servo marks wherein one of the servo marks is an INDEX mark, wherein the media is rotated within the magnetic media system in relation to a read/write head for writing a sector to or reading a sector from one of a second plurality of information regions, each region being positioned along one of the tracks between two adjacent servo marks such that each region has a capacity for storing a third plurality of information bits and wherein each sector includes a fourth plurality of information bits, the method having a capability to start an operation at one of the servo marks without waiting for an INDEX mark and comprising the steps of:
 - 15 a. detecting a number of a current servo mark representing the read/write head's position relative to the media;
 - 16 b. initializing a target servo mark to equal a first servo mark for which the system will begin operation, the target servo mark initialized to a servo mark which is a predetermined number of servo marks ahead of the current servo mark;
 - 17 c. converting the target servo mark to a relative target servo mark within a frame, the frame including N sectors and M servo marks;
 - 18 d. incrementing the current servo mark as a next servo mark is detected;
 - 19 e. fetching an entry from a frame table corresponding to the relative target servo mark, the entry representing the split field and number of sector pulses to be

70 automatically generated after the target servo mark,
31 while waiting until the current servo mark is equal to
32 the target servo mark;
33 f. incrementing the target servo mark as each next servo
34 mark is detected, once the current servo mark equals
35 the target servo mark, forming an incremented target
36 servo mark;
37 g. converting the incremented target servo mark to an
38 incremented relative target servo mark within a frame;
39 h. fetching an entry from the frame table corresponding to
40 the incremented relative target servo mark, wherein
41 after every N sectors and M servo marks, the frame
42 repeats itself; and
43 i. repeating steps f-h until the operation is complete.

1 2. The method as claimed in claim 1 wherein a last frame
2 on a track can be a partial frame.

1 3. The method as claimed in claim 1 wherein each sector
2 comprises a plurality of fields, each of which can be split so
3 that a first portion of the sector is stored before a servo mark
4 and a second remaining portion of the sector is stored after a
5 servo mark.

1 4. The method as claimed in claim 1 wherein each sector
2 comprises a plurality of fields, each of which can be split so
3 that a first portion of the sector is stored before a servo mark
4 and the remaining portion of the sector can be stored after one
5 or more servo marks.

1 5. The method as claimed in claim 1 further comprising the
2 step of automatically generating a number of sector pulses after
3 each servo mark corresponding to the entry in the frame table.

4 6. The method as claimed in claim 1 wherein one frame
5 equals a track length.

1 7. The method as claimed in claim 3 further comprising the
2 step of loading the fetched entry from the frame table into
3 internal registers and counters within the system once the fetch
4 of an entry from a frame table has been completed.

1 8. The method as claimed in claim 7 wherein the entry from
2 the frame table includes a sector pulse count, a delay value, a
3 split field count value and split field flags indicating which
4 field has been split.

1 9. The method as claimed in claim 8 wherein the first
2 delay value represents a first delay time period after a servo
3 mark to a generation of a first sector pulse for a non-zero
4 sector pulse count.

1 10. The method as claimed in claim 9 wherein the split
2 field count value represents a number of bytes within a split
3 sector field which occurs before assertion of a next servo mark.

1 11. The method as claimed in claim 10 wherein the frame
2 table is stored in buffer random access memory of the system.

1 12. An apparatus for automatic split field calculation in a
2 magnetic media system using a magnetic media having a first
3 plurality of circular tracks, each track having a second
4 plurality of radial servo marks wherein one of the servo marks is
5 an INDEX mark, wherein the media is rotated within the magnetic
6 media system in relation to a read/write head for writing a
7 sector to or reading a sector from one of a second plurality of
8 information regions, each region being positioned along one of

9 the tracks between two adjacent servo marks such that each region
10 has a capacity for storing a third plurality of information bits
11 and wherein each sector includes a fourth plurality of
12 information bits, the apparatus having a capability to start an
13 operation at one of the servo marks without waiting for an INDEX
14 mark and comprising:

- 15 a. a detecting circuit for detecting a number of a current
16 servo mark representing the read/write head's position
17 relative to the media;
- 18 b. an initializing circuit coupled to the detecting
19 circuit for initializing a target servo mark to equal a
20 first servo mark for which the system will begin
21 operation, the target servo mark initialized to a servo
22 mark which is a predetermined number of servo marks
23 ahead of the current servo mark;
- 24 c. a converting circuit coupled to the initializing
25 circuit for converting the target servo mark to a
26 relative target servo mark within a frame, the frame
27 including N sector and M servo marks;
- 28 d. a first incrementing circuit coupled to the converting
29 circuit for incrementing the current servo mark as a
30 next servo mark is detected;
- 31 e. a second incrementing circuit coupled to the first
32 incrementing circuit and the converting circuit for
33 incrementing the target servo mark as each servo mark
34 is detected, once the current servo mark is equal to
35 the target servo mark, forming an incremented target
36 servo mark;
- 37 f. a frame table coupled to the converting circuit and to
38 the incrementing circuit for storing a fifth plurality

39 of entries, each of the fifth plurality of entries
40 representing a split field and number of sector pulses
41 to be automatically generated after the target servo
42 mark;

43 g. a retrieving circuit coupled to the frame table for
44 fetching an entry from the frame table corresponding to
45 the relative target servo mark, while waiting until the
46 current servo mark is equal to the target servo mark
47 and subsequently fetching corresponding subsequent
48 entries from the frame table for each subsequent servo
49 mark after the target servo mark corresponding to the
50 incremented target servo mark and a relative servo mark
51 within the frame, wherein after every N sectors and M
52 servo marks, the frame repeats itself; and
53 h. a generating circuit coupled to the retrieving circuit
54 and to the operating circuit for automatically
55 generating sector pulses.

1 13. The apparatus as claimed in claim 12 wherein a last
2 frame on a track can be a partial frame.

1 14. The apparatus as claimed in claim 12 wherein one frame
2 equals a track length.

1 15. The apparatus as claimed in claim 12 wherein each
2 sector comprises a plurality of fields, each of which can be
3 split so that a first portion of the sector is stored before a
4 servo mark and a second remaining portion of the sector is stored
 after a servo mark.

1 16. The apparatus as claimed in claim 12 wherein each
2 sector comprises a plurality of fields, each of which can be
3 split so that a first portion of the sector is stored before a
4 servo mark and the remaining portion of the sector can be stored
5 after one or more servo marks.

1 17. The apparatus as claimed in claim 15 further comprising
2 a sixth plurality of internal registers and counters coupled to
3 the retrieval circuit for storing the fetched entry from the
4 frame table.

1 18. The apparatus as claimed in claim 17 wherein each entry
2 from the frame table includes a sector pulse count, a first delay
3 value, a split field count value and a seventh plurality of split
4 field flags indicating which field has been split.

1 19. The apparatus as claimed in claim 18 wherein the frame
2 table is stored in buffer random access memory of the system.

1 20. A table driven method of automatic split field
2 calculation in magnetic media systems using media which have a
3 first plurality of circular tracks, each track having an INDEX
4 mark, and a second plurality of radial servo marks, wherein the
5 media is rotated within the magnetic media system in relation to
6 a read/write head, the method having the capability to start from
7 any servo mark on the track without waiting for an INDEX mark and
8 comprising the steps of:

- 9 a. detecting a number of a current servo mark representing
10 the read/write head's position relative to the media;
- 11 b. initializing a target servo mark count to equal a first
12 servo mark for which the system will begin operation,

the target servo mark initialized to a servo mark which is a predetermined number of servo marks ahead of the current servo mark;

- c. converting the target servo mark to a relative target servo mark within a frame, the frame including N sectors and M servo marks;
- d. fetching an entry from a frame table representing the relative target servo mark;
- e. comparing the target servo mark to the current servo mark;
- f. incrementing the current servo mark as a next servo mark is detected;
- g. repeating steps d-f until the target servo mark is equal to the current servo mark;
- h. repeating steps a-g if the fetch of an entry from a frame table has not yet been completed;
- i. loading the entry from the frame table into internal registers within the system once the fetch of an entry from a frame table has been completed;
- j. providing split field information for each sector pulse from the current servo mark to the next servo mark;
- k. waiting a first delay time period before generating a first sector pulse after the current servo mark or skipping steps k-m if no sector pulse is generated before the next servo mark;
- l. generating a predetermined number of sector pulses after the first sector pulse and before a next servo mark;
- m. waiting a second delay time period from a last sector pulse generated until the assertion of the next servo mark;
- n. incrementing the relative target servo mark after the second delay time period for the next servo mark; and

46 o. repeating steps h-1 for each subsequent servo mark
47 until the operation is complete.

1 21. The method as claimed in claim 20 wherein the entry
2 from the frame table includes a sector pulse count, a first delay
3 value, a split field count value and a third plurality of split
4 flags for each servo mark in the frame.

1 22. The method as claimed in claim 21 wherein the sector
2 pulse count represents the predetermined number of sector pulses
3 plus the first sector pulse for a non-zero sector pulse count.

1 23. The method as claimed in claim 22 wherein the first
2 delay value represents the first delay time period.

1 24. The method as claimed in claim 23 wherein the split
2 field count value corresponds to the second delay time period and
3 represents a number of bytes within a split sector field which
4 occurs before assertion of a next servo mark.

1 25. The method as claimed in claim 20 wherein each sector
2 comprises a fourth plurality of fields, each of which can be
3 split so that a first portion of the sector is stored before a
4 servo mark and a second remaining portion of the sector is stored
5 after a servo mark.

1 26. The method as claimed in claim 20 wherein each sector
2 comprises a plurality of fields, each of which can be split so
3 that a first portion of the sector is stored before a servo mark
4 and the remaining portion of the sector can be stored after one
5 or more servo marks.

1. 27. An apparatus for automatic split field calculation in
2 magnetic media systems using media which have a first plurality
3 of circular tracks, each track having an INDEX mark, and a second
4 plurality of radial servo marks, wherein the media is rotated
5 within the magnetic media system in relation to a read/write
6 head, the apparatus having the capability to start from any servo
7 mark on the track without waiting for an INDEX mark and
8 comprising:

9. a. a detecting circuit for detecting a current servo mark
10 corresponding to the position of the read/write head;
11. b. a target servo mark counter coupled to the detecting
12 circuit for storing a first servo mark for which the
13 system will begin operation, the target servo mark
14 initialized to a servo mark which is a predetermined
15 number of servo marks ahead of the current servo mark;
16. c. a frame table comprising a third plurality of entries
17 coupled to the target servo mark counter, each entry
18 representing split field and sector pulse information
19 for a servo mark within the frame;
20. d. a comparator coupled to the target servo mark counter
21 and to the detecting circuit, for comparing the target
22 servo mark counter to the current servo mark; and
23. e. an internal register network coupled to the frame table
24 for storing split field and sector pulse information
25 from the frame table for the target servo mark.

1. 28. The apparatus as claimed in claim 27 wherein each of
2 the third plurality of entries include a sector pulse count
3 value, a first delay value, a split field count value and a
4 fourth plurality of split flags corresponding to a servo mark
5 within the frame.

1 29. The apparatus as claimed in claim 28 wherein the sector
2 pulse count value represents a number of sector pulses to be
3 generated after the corresponding servo mark and before a next
4 servo mark.

1 30. The apparatus as claimed in claim 29 wherein the first
2 delay value represents a first delay period corresponding to a
3 delay from assertion of the current servo mark until an assertion
4 of a first sector pulse for a non-zero sector pulse count value.

1 31. The apparatus as claimed in claim 30 wherein the split
2 field count value represents a number of bytes within a split
3 sector field which occurs before assertion of a next servo mark.

1 32. The apparatus as claimed in claim 27 wherein each
2 sector comprises a fifth plurality of fields, each of which can
3 be split so that a first portion of the sector is stored before a
4 servo mark and a second remaining portion of the sector is stored
after a servo mark.

1 33. The apparatus as claimed in claim 27 wherein each
2 sector comprises a fifth plurality of fields, each of which can
3 be split so that a first portion of the sector is stored before a
4 servo mark and the remaining portion of the sector can be stored
5 after one or more servo marks.

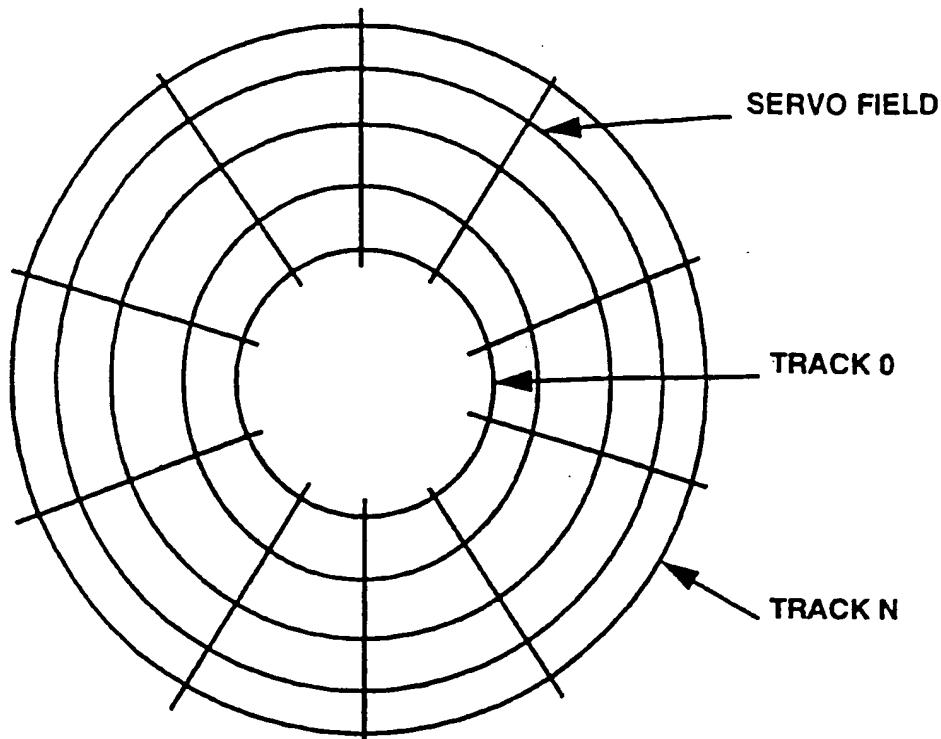


FIGURE 1
(PRIOR ART)

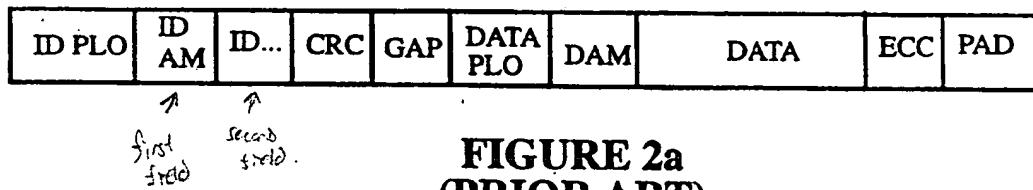


FIGURE 2a
(PRIOR ART)

PLO	SYNC	HEADER	DATA	ECC	PAD
-----	------	--------	------	-----	-----

FIGURE 2b

1/5
 RECTIFIED SHEET (RULE 91)
 ISA/EP

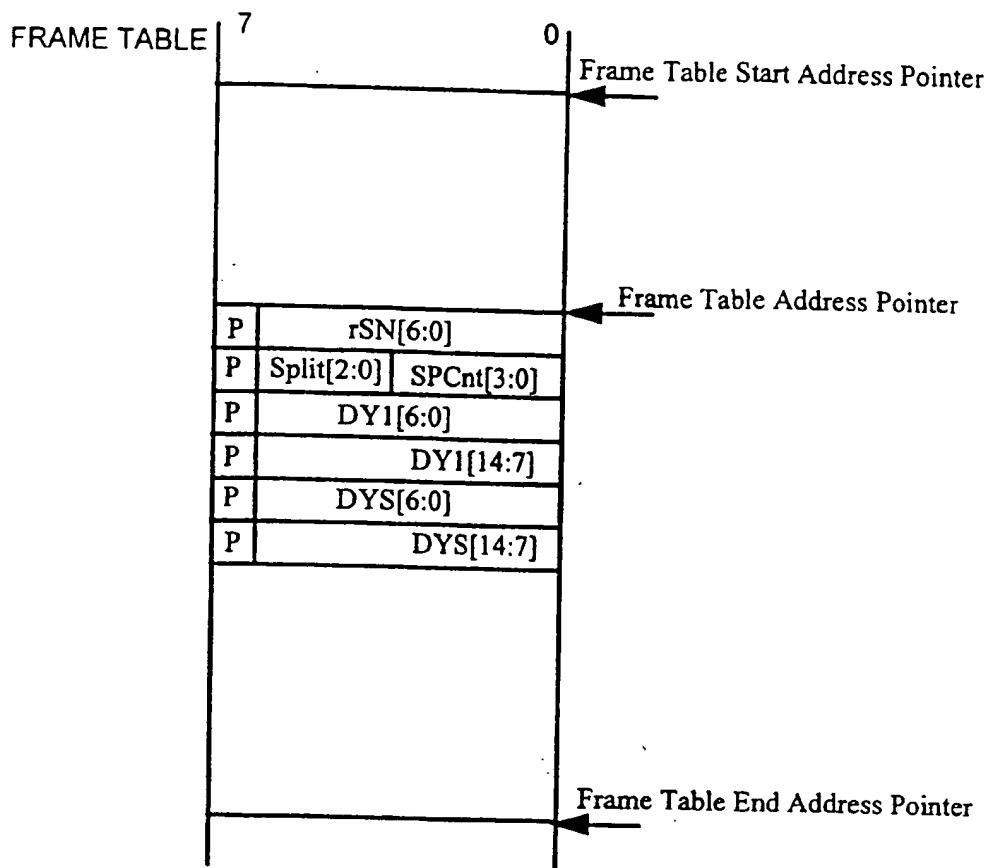


FIGURE 3

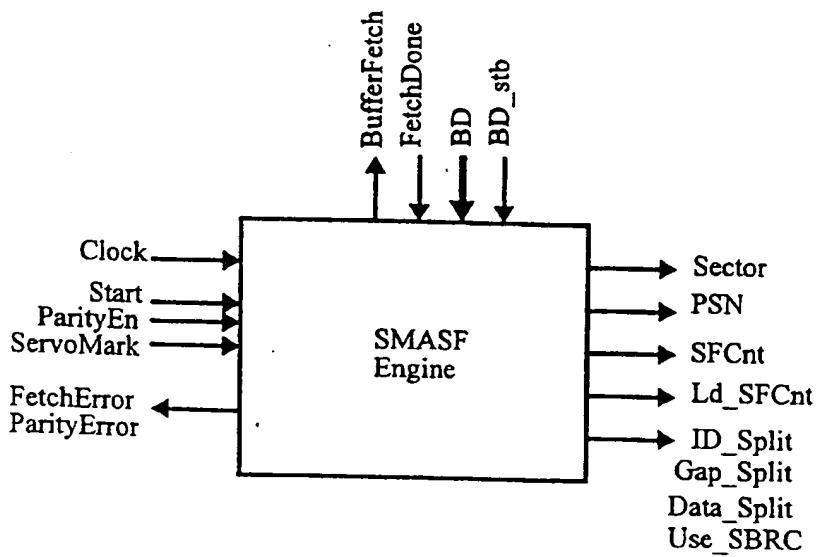


FIGURE 4

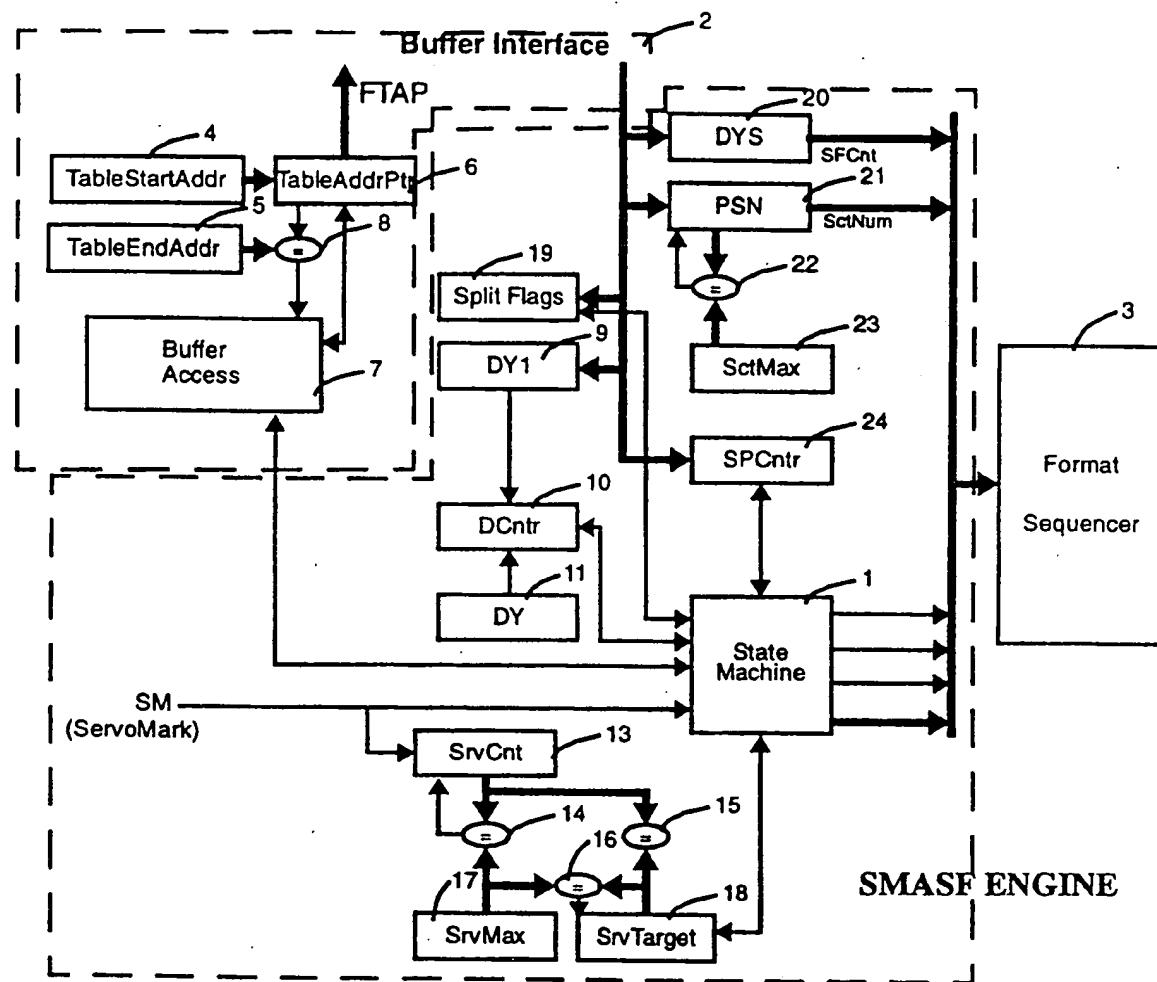


FIGURE 5

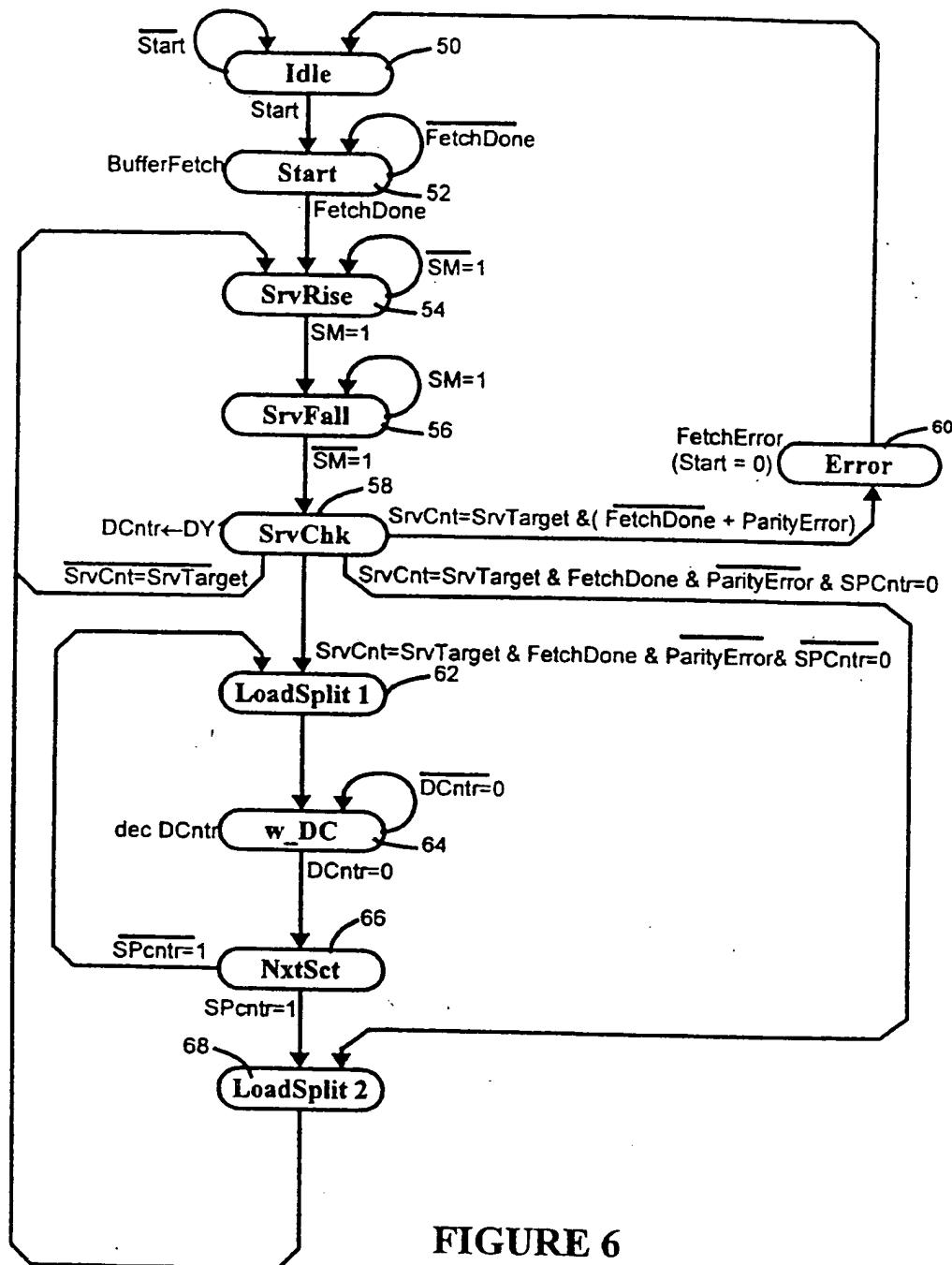


FIGURE 6

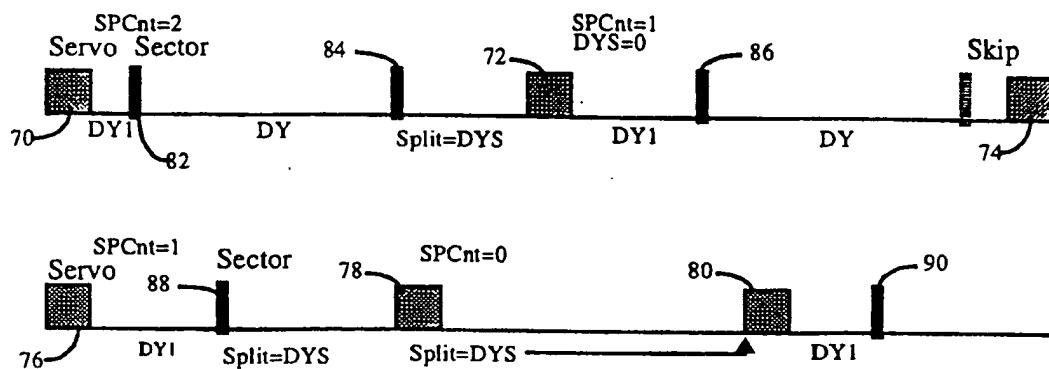


FIGURE 7

INTERNATIONAL SEARCH REPORT

Initial Application No.
PCT/US 95/02608A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G11B20/12 G11B20/10 G11B5/596 G11B27/10

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G11B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO-A-90 15407 (MAXTOR CORPORATION) 13 December 1990 see page 11, line 5 - page 13, line 18 see page 20, line 19 - page 21, line 27 see page 25, line 8 - page 26, line 15 ---	1, 12, 20, 27
A	EP-A-0 565 856 (HEWLETT-PACKARD COMPANY) 20 October 1993 see column 4, line 18 - line 53; figure 2 see column 6, line 21 - column 7, line 50 ---	1, 5, 9, 12, 20, 27
A	BYTE, vol. 17, no. 3, March 1992 pages 195-203, ROD KIRK ET AL. 'Embedded Intelligence' see page 196, right column, last paragraph - page 198, right column, paragraph 3; figure 2 ---	1, 3, 4, 12, 20, 27

 Further documents are listed in the continuation of box C. Patent family members are listed in annex.

* Special categories of cited documents :

- 'A' document defining the general state of the art which is not considered to be of particular relevance
- 'E' earlier document but published on or after the international filing date
- 'L' document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- 'O' document referring to an oral disclosure, use, exhibition or other means
- 'P' document published prior to the international filing date but later than the priority date claimed

- 'T' later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- 'X' document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- 'Y' document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- '&' document member of the same patent family

Date of the actual completion of the international search

29 June 1995

Date of mailing of the international search report

07.07.95

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl
Fax (+ 31-70) 340-3016

Authorized officer

Brunet, L

INTERNATIONAL SEARCH REPORT

Intell Application No
PCT/US 95/02608

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP-A-0 497 593 (QUANTUM CORPORATION) 5 August 1992 cited in the application see page 9, line 57 - page 10, line 24 ---	1,12,20, 27
A	EP-A-0 519 764 (DIGITAL EQUIPMENT CORPORATION) 23 December 1992 see column 8, line 6 - column 9, line 16 ---	1,12,20, 27
A	EP-A-0 517 478 (QUANTUM CORPORATION) 9 December 1992 ---	
A	US-A-4 669 004 (MOON ET AL.) 26 May 1987 -----	

3

INTERNATIONAL SEARCH REPORT

Int. Application No.
PCT/US 95/02608

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
WO-A-9015407	13-12-90	US-A-	4999720	12-03-91
		AU-A-	6075390	07-01-91
		CA-A-	2060197	06-12-90
		EP-A-	0476067	25-03-92
		JP-T-	4507474	24-12-92
EP-A-565856	20-10-93	US-A-	5276564	04-01-94
		JP-A-	6020393	28-01-94
EP-A-497593	05-08-92	US-A-	5241546	31-08-93
		JP-A-	5347075	27-12-93
		US-A-	5255136	19-10-93
EP-A-519764	23-12-92	US-A-	5278703	11-01-94
		JP-A-	5234256	10-09-93
EP-A-517478	09-12-92	JP-A-	6124554	06-05-94
		US-A-	5420730	30-05-95
US-A-4669004	26-05-87	US-A-	4783705	08-11-88

This Page is inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ~~BLACK BORDERS~~
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT OR DRAWING
- BLURED OR ILLEGIBLE TEXT OR DRAWING
- SKEWED/SLANTED IMAGES
- COLORED OR BLACK AND WHITE PHOTOGRAPHS
- GRAY SCALE DOCUMENTS
- LINES OR MARKS ON ORIGINAL DOCUMENT
- REPERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning documents *will not* correct images
problems checked, please do not report the
problems to the IFW Image Problem Mailbox